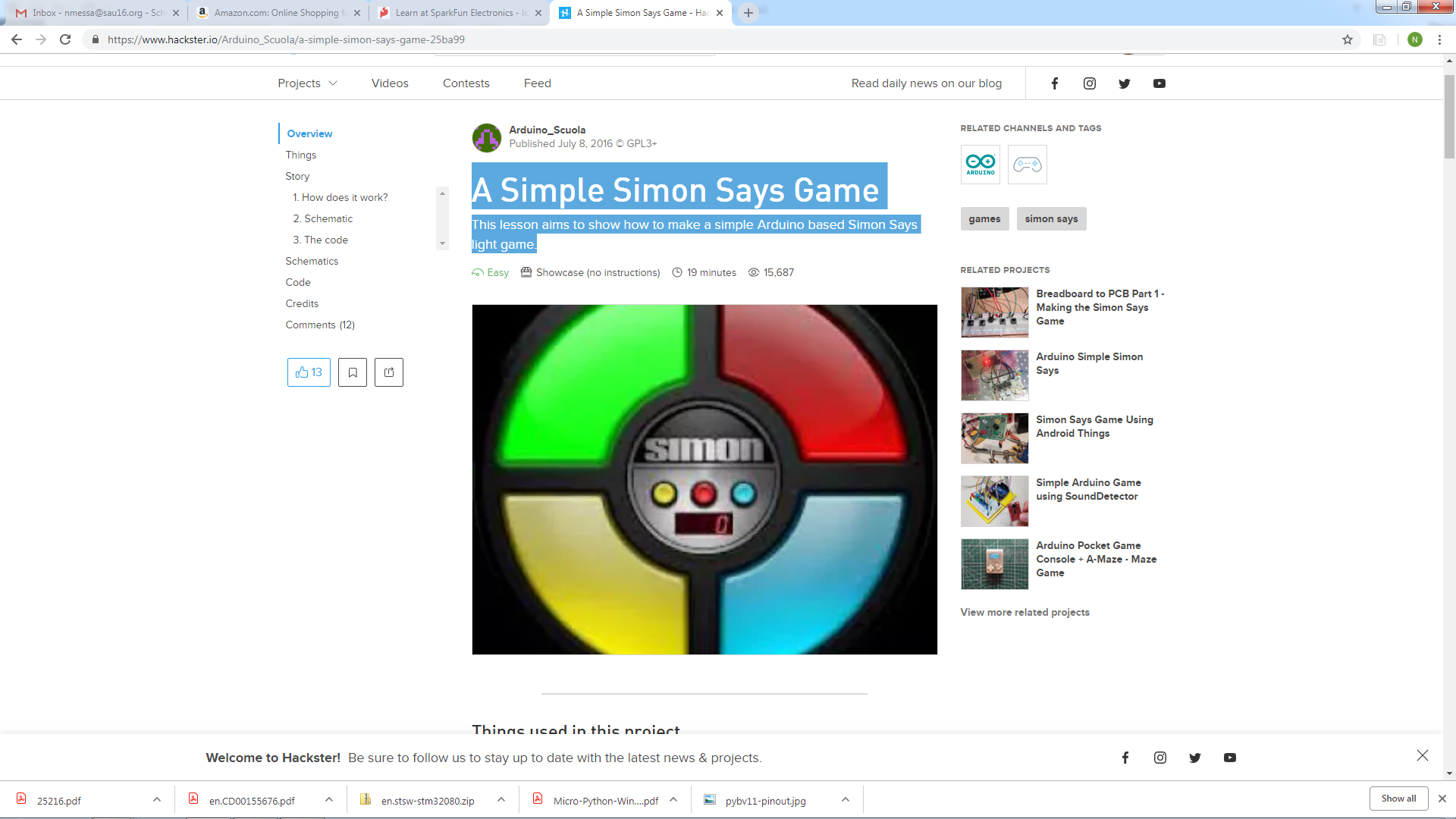
A Simple Simon Says Game

This lesson aims to show how to make a simple Arduino based Simon Says light game.



|  |
| --- |
|  |
| Hardware components | | | | | |
| Arduino UNO & Genuino UNO | |  | | --- | | [Arduino UNO & Genuino UNO](https://www.hackster.io/arduino/products/arduino-uno-genuino-uno) | |  | | × | 1 |  |  |
|  | |  | | --- | | Breadboard | |  | | × | 1 |  |  |
|  | |  | | --- | | Red Led 5mm | |  | | × | 1 |  |  |
|  | |  | | --- | | Yellow Led 5mm | |  | | × | 1 |  |  |
|  | |  | | --- | | Green Led 5mm | |  | | × | 1 |  |  |
|  | |  | | --- | | Blue Led 5mm | |  | | × | 1 |  |  |
|  | |  | | --- | | PushButton 6x6 | |  | | × | 1 |  |  |
|  | |  | | --- | | 10 jumper wires 100mm male to male | |  | | × | 1 |  |  |

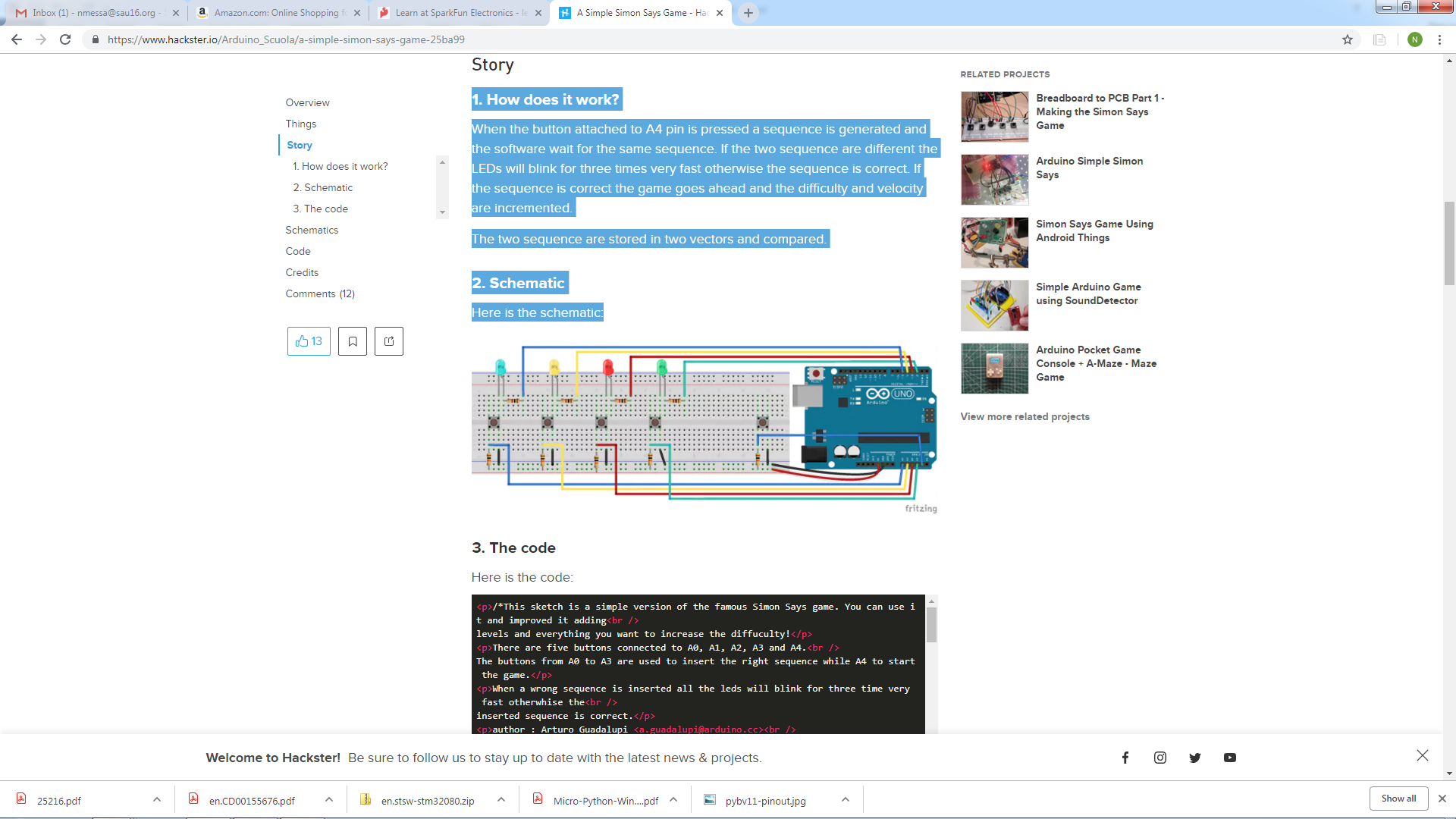
### 1. How does it work?

When the button attached to A4 pin is pressed a sequence is generated and the software wait for the same sequence. If the two sequence are different the LEDs will blink for three times very fast otherwise the sequence is correct. If the sequence is correct the game goes ahead and the difficulty and velocity are incremented.

The two sequence are stored in two vectors and compared.

### 2. Schematic

Here is the diagram:



/\*This sketch is a simple version of the famous Simon game.

There are five buttons connected to A0, A1, A2, A3 and A4. The buttons from A0 to A3 are used to insert the right sequence while A4 to start the game.

When a wrong sequence is inserted all the LED’s will blink for three times very fast otherwise the inserted sequence is correct.

Hardware needed:

5x pushbuttons

1x Blue led

1x Yellow led

1x Red led

1x Green Led

4x 1k resistors

4x 10k resisors

10x jumpers

\*/

**3. The Code**

const int MAX\_LEVEL = 100;

int sequence[MAX\_LEVEL];

int your\_sequence[MAX\_LEVEL];

int level = 1;

int velocity = 1000;

void setup() {

pinMode(A0, INPUT);

pinMode(A1, INPUT);

pinMode(A2, INPUT);

pinMode(A3, INPUT);

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

}

void loop()

{

if (level == 1)

generate\_sequence();//generate a sequence;

//If start button is pressed or you're winning

if (digitalRead(A4) == LOW || level != 1)

{

show\_sequence(); //show the sequence

get\_sequence(); //wait for your sequence

}

}

void show\_sequence()

{

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

for (int i = 0; i < level; i++)

{

digitalWrite(sequence[i], HIGH);

delay(velocity);

digitalWrite(sequence[i], LOW);

delay(200);

}

}

void get\_sequence()

{

int flag = 0; //this flag indicates if the sequence is correct

for (int i = 0; i < level; i++)

{

flag = 0;

while(flag == 0)

{

if (digitalRead(A0) == LOW)

{

digitalWrite(5, HIGH);

your\_sequence[i] = 5;

flag = 1;

delay(200);

if (your\_sequence[i] != sequence[i])

{

wrong\_sequence();

return;

}

digitalWrite(5, LOW);

}

if (digitalRead(A1) == LOW)

{

digitalWrite(4, HIGH);

your\_sequence[i] = 4;

flag = 1;

delay(200);

if (your\_sequence[i] != sequence[i])

{

wrong\_sequence();

return;

}

digitalWrite(4, LOW);

}

if (digitalRead(A2) == LOW)

{

digitalWrite(3, HIGH);

your\_sequence[i] = 3;

flag = 1;

delay(200);

if (your\_sequence[i] != sequence[i])

{

wrong\_sequence();

return;

}

digitalWrite(3, LOW);

}

if (digitalRead(A3) == LOW)

{

digitalWrite(2, HIGH);

your\_sequence[i] = 2;

flag = 1;

delay(200);

if (your\_sequence[i] != sequence[i])

{

wrong\_sequence();

return;

}

digitalWrite(2, LOW);

}

}//end of while loop

}//end of for loop

right\_sequence();

}

void generate\_sequence()

{

randomSeed(millis()); //in this way is really random!!!

for (int i = 0; i < MAX\_LEVEL; i++)

{

sequence[i] = random(2,6);

}

}

void wrong\_sequence()

{

for (int i = 0; i < 3; i++)

{

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, HIGH);

delay(250);

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

delay(250);

}

level = 1;

velocity = 1000;

}

void right\_sequence()

{

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

delay(250);

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, HIGH);

delay(500);

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

delay(500);

if (level < MAX\_LEVEL); //Do Nothing

level++;

velocity -= 50; //increase difficulty

}